

CLAIMS

1. A system for learning model-based lifecycle diagnostics, the system comprising:
an integrated development environment having software tools linked within;
a run-time environment having agents that detect failures linked within; and
a bi-directional link between the integrated development environment and the
run-time environment;

whereby the failures detected in the run-time environment are traced back to the
integrated development environment to determine model errors.
2. A system according to claim 1, wherein the integrated development environment
includes requirements management tools, design tools, and implementation tools linked
together.
3. A system according to claim 2, wherein the requirements management tools
includes an object oriented requirements management tool and an issue-based
information system requirements management tool.
4. A system according to claim 2, wherein the design tools include an object
oriented model driven function design tool, a knowledge-based diagnostics design tool,
and a model-based diagnostic design tool.
5. A system according to claim 2, wherein the implementation tools include a
software function code generation, management, and deployment tool, and a software
diagnostic code generation, management.
6. A system according to claim 1, wherein the run-time environment includes
diagnostic agents.

7. A system according to claim 6, wherein the diagnostic agents include model-based diagnostic agents and learning model-based diagnostic agents.
8. A system according to claim 1, wherein the run-time environment includes a database, a server software tool, a broker, and diagnostic agents.
9. A system according to claim 1, wherein the bi-directional link is a DRD link.
10. A system according to claim 9, wherein the DRD link includes a database.
11. A system according to claim 10, wherein the database is a distributed database.
12. A system for learning model-based lifecycle diagnostics for vehicles, the system comprising:
 - an integrated development environment including a requirements management tool, a design tool, and a deployment tool linked together;
 - a run-time environment having applications, brokers, and agents that detect failures linked together; and
 - a bi-directional link between the integrated development environment and the run-time environment;
 - whereby the failures detected in the run-time environment are traced back to the integrated development environment to determine model errors.
13. A system according to claim 12, wherein the requirements management tool includes an object oriented requirements management tool and an issue-based information system requirements management tool.
14. A system according to claim 12, wherein the design tool includes an object oriented model driven function design tool, a knowledge-based diagnostics design tool, and a model-based diagnostic design tool.

15. A system according to claim 12, wherein the implementation tool include a software function code generation, management, and deployment tool, and a software diagnostic code generation, management.
16. A system according to claim 12, wherein the agents are diagnostic agents.
17. A system according to claim 16, wherein the diagnostic agents include model-based diagnostic agents and learning model-based diagnostic agents.
18. A system according to claim 12, wherein the bi-directional link is a DRD link.
19. A system according to claim 18, wherein the DRD link includes a database.
20. A system according to claim 19, wherein the database is a distributed database.
21. A method of diagnosing model errors in a software environment including an integrated development environment and a run-time environment bi-directionally linked by a link, the method comprising:
 - detecting failures within the run-time environment;
 - tracing the failures back to the integrated development environment; and
 - identifying the model errors in the integrated development environment based on the tracing of the failures.
22. A method according to claim 21, wherein detecting failures includes using model-based diagnostic agents to detect failures within the run-time environment.
23. A method according to claim 22, further comprising determining root causes for known failure modes based on the failures detected by the model-based diagnostic agents.
24. A method according to claim 21, wherein detecting failures includes using learning model-based diagnostic agents to detect failures within the run-time

environment.

25. A method according to claim 24, wherein detecting failures includes the diagnostic agents using embedded data mining algorithms that learn a model by observing.

26. A method according to claim 24, wherein tracing failures includes the diagnostic agents writing information into the link.

27. A method according to claim 26, wherein tracing failures includes reading the information in the link into the integrated development environment.

28. A method according to claim 27, wherein identifying the model errors includes identifying the model errors in the levels of the model represented at the levels of implementation, design, and requirements.

29. A computer program product readable by a computing system and encoding instructions diagnosing model errors in a software environment including an integrated development environment and a run-time environment bi-directionally linked, the computer process comprising:

detecting failures within the run-time environment;

tracing the failures back to the integrated development environment; and

identifying the model errors in the integrated development environment based on the tracing of the failures.

30. A computer program product according to claim 29, wherein detecting failures includes using model-based diagnostic agents to detect failures within the run-time environment.

31. A computer program product according to claim 30, further comprising determining root causes for known failure modes based on the failures detected by the model-based diagnostic agents.
32. A computer program product according to claim 29, wherein detecting failures includes using learning model-based diagnostic agents to detect failures within the run-time environment.
33. A computer program product according to claim 32, wherein detecting failures includes the diagnostic agents using embedded data mining algorithms that learn a model by observing.
34. A computer program product according to claim 32, wherein tracing failures includes the diagnostic agents writing information into the link.
35. A computer program product according to claim 34, wherein tracing failures includes reading the information in the link into the integrated development environment.
36. A computer program product according to claim 35, wherein identifying the model errors includes identifying the model errors in the levels of the model represented at the levels of implementation, design, and requirements.